Дніпровський ліцей інформаційних технологій

при Дніпровському національному університеті

імені Олеся Гончара

Випускна робота

**Моделювання еволюції колонії мікроорганізмів на площині за умов неперервної зміни умов виживання окремих мікроорганізмів**

Виконавець:

Ліцеїст 11 – Ф – 1 класу

Москаленко Сергій

Науковий керівник:

Ентін Й. А

Дніпро

2018

**ЗМІСТ:**

[ВСТУП 3](#_Toc530934817)

[ОСНОВНА ЧАСТИНА 4](#_Toc530934818)

[РОЗДІЛ1 Налаштування 4](#_Toc530934819)

[РОЗДІЛ2 Графіки 9](#_Toc530934820)

[РОЗДІЛ3 Стандартна колонія мікроорганізмів 12](#_Toc530934821)

[РОЗДІЛ4 Колонія мікроорганізмів з часовими полями 14](#_Toc530934822)

[РОЗДІЛ5 Колонія мікроорганізмів з різною частотою змін умов та зміни покоління 16](#_Toc530934823)

[ВИСНОВКИ 18](#_Toc530934824)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 19](#_Toc530934825)

[УМОВНІ ПОЗНАЧЕННЯ 19](#_Toc530934826)

ВСТУП

**Формулювання задачі.**

Є прямокутник, заповнений квадратами. У кожному може бути мікроорганізм, який виживає і може розмножуватися, якщо в нього не більше від N1 і не менше від N2 сусідів. Початкову кількість мікроорганізмів у прямокутнику, а також інтервал часу між тактами (змінами стану колонії мікроорганізмів) задаємо. Та далі умови виживання коливаються. Треба дослідити вплив таких змін на еволюцію колонії мікроорганізм.

**Мета**

Дослідити зв’язок між частотою змін умов виживання окремих М та існуванням КМ.

**Актуальність**

Моделювання впливу коливань зовнішніх умов на стан КМ є корисним для боротьби зі шкідливими вірусами.

ОСНОВНА ЧАСТИНА

Всього в проекті є три практичні частини. В кожній із частин є такі кнопки, як «Розпочати», «Налаштування» і «Графіки». Також на кожній формі

РОЗДІЛ1 Налаштування

Щоб передати нові налаштування з однієї форми на іншу я використовував змінні(на формі, де є КМ) та функції(на формі налаштувань).

Всі змінні, що використовуються для налаштувань у кожній формі:

//змінні, що відповідають за налаштування:

bool a1;//true: виживання по кількості

//false: виживання по розташуванню

bool[] v;

int vmin;

int vmax;

bool b1;//true: народження по кількості

//false: народження по розташуванню

bool[] b;

int bmin;

int bmax;

int size\_micr;

bool diff\_cycles;

**1.Вибір умов виживання.**

Якщо вибрано виживання в залежності від кількості сусідів, то стає доступною настройка «М залишаються у живих, якщо кількість сусідів буде коливатися в інтервалі».

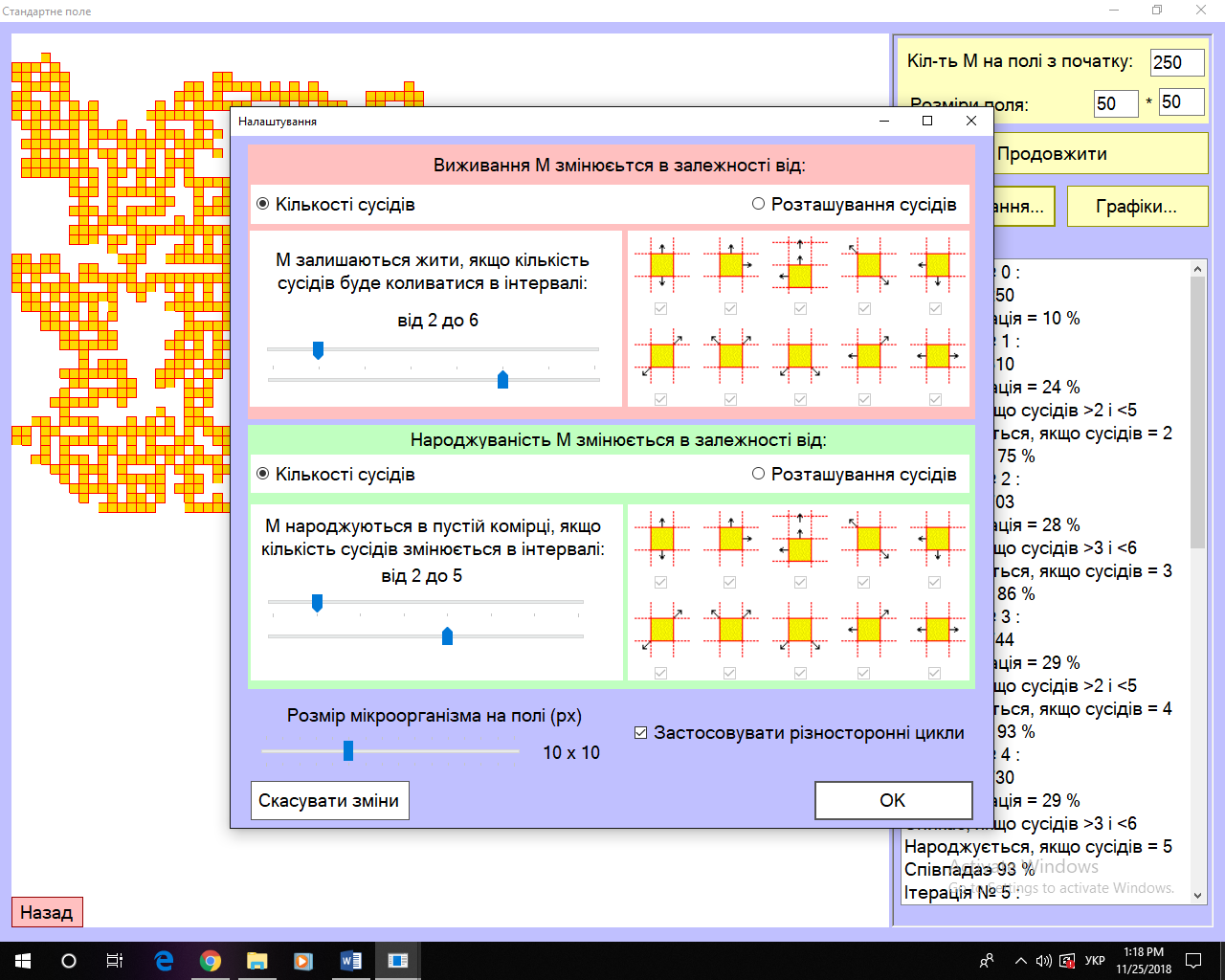


Рисунок 1. Виживання мікроорганізмів в залежності від кількості сусідів

Це означає, що при кожній ітерації проміжок кількості сусідів буде змінюватися у відведених межах. Наприклад, якщо на верхньому trackBar - значення, показане на рис. то мінімальна умова виживання коливається від 2 до середнього арифметичного між максимальним і мінімальним значеннями, тобто до 3, а максимальна - від 5 до 6 включно. В даному випадку М виживе, якщо кількість сусідів більше [2; 4) і менше ніж (4; 6];

Якщо виживання залежить від розташування сусідів(рис.2.), то доступне налаштування, показане на рис.

Тут все очевидно: виділені жовтим checkBox, по черзі перевіряються;

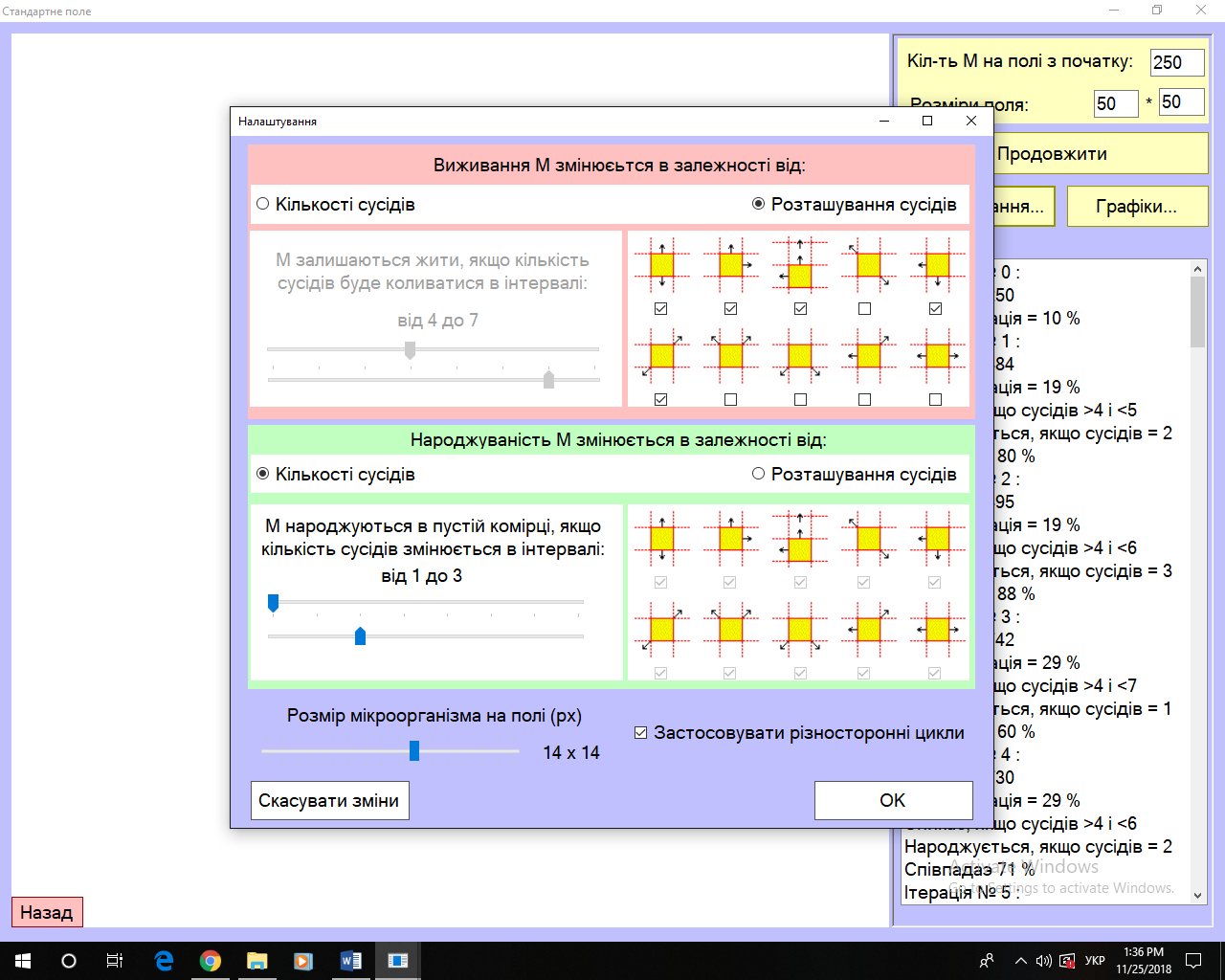


Рисунок 2 Виживання в залежності від розташування мікроорганізмів

У коді для виключення невибраних комбінацій є обмежувач:

int id\_limit(int id, bool p)

{//змінна 'id' - послідовний номер комбінації

repeat:

if (id == 1 && ((v[0] == false && p == false) || (b[0] == false && p == true)))

id++;

if (id == 2 && ((v[1] == false && p == false) || (b[1] == false && p == true)))

id++;

if (id == 3 && ((v[2] == false && p == false) || (b[2] == false && p == true)))

id++;

if (id == 4 && ((v[3] == false && p == false) || (b[3] == false && p == true)))

id++;

if (id == 5 && ((v[4] == false && p == false) || (b[4] == false && p == true)))

id++;

if (id == 6 && ((v[5] == false && p == false) || (b[5] == false && p == true)))

id++;

if (id == 7 && ((v[6] == false && p == false) || (b[6] == false && p == true)))

id++;

if (id == 8 && ((v[7] == false && p == false) || (b[7] == false && p == true)))

id++;

if (id == 9 && ((v[8] == false && p == false) || (b[8] == false && p == true)))

id++;

if (id == 10 && ((v[9] == false && p == false) || (b[9] == false && p == true)))

{

id = 1;

goto repeat;

}

return id;

}

**2.Налаштування народження.**

Кількість сусідів для народження на порожньому місці (рис.3.), по черзі змінюється (в даному випадку від 2 до 5 включно);

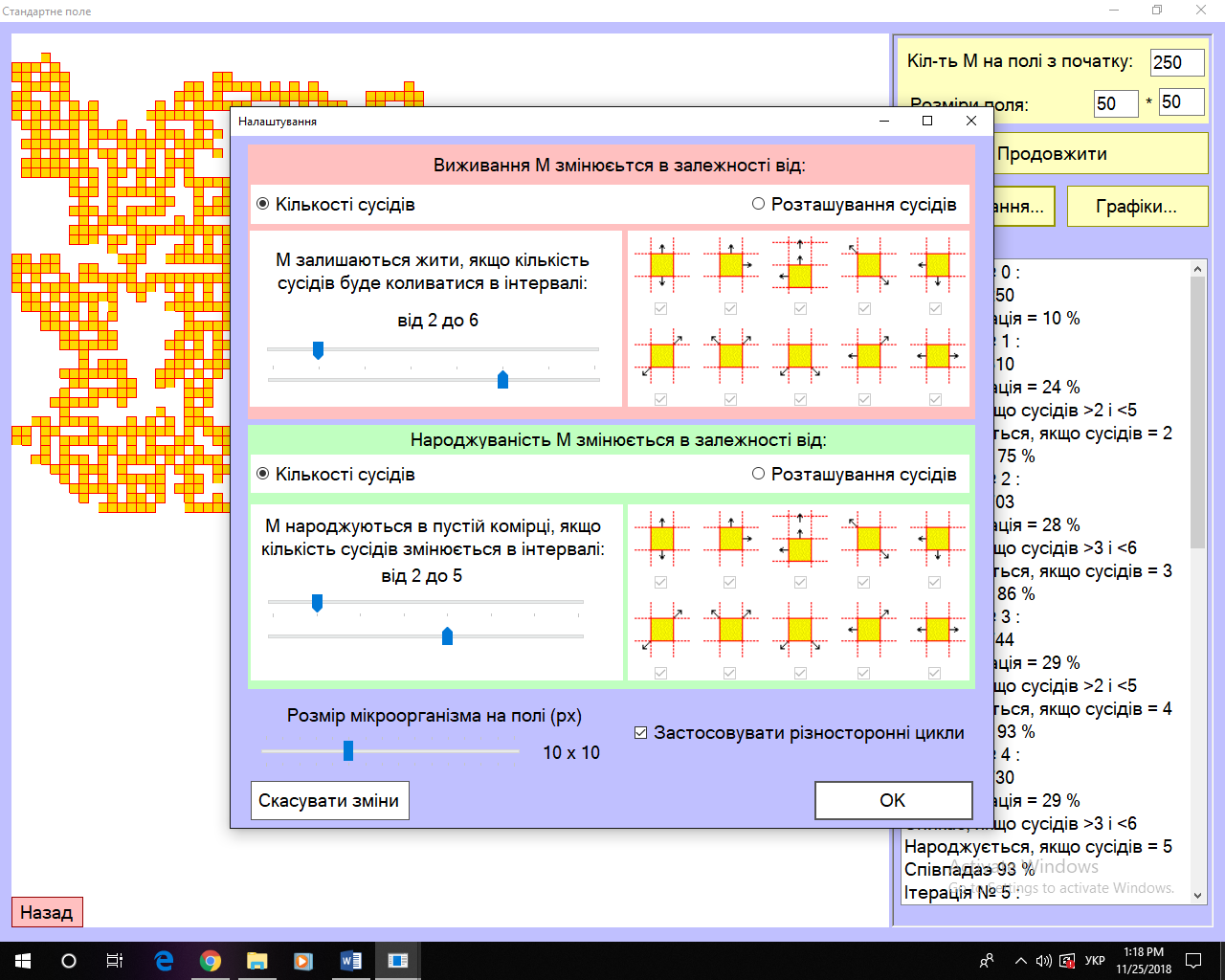


Рисунок 3. Налаштування народження в залежності від кількості

**3.Різносторонні цикли.**

Якщо це налаштування активоване, то всі перевірки на виживання і народження М будуть перевірятися з-під різних кутів в наступній послідовності:

1. по рядках зліва на право і зверху вниз,
2. за стовпцями з зверху вниз і справа наліво,
3. по рядках з права на ліво і від низу до верху,
4. за стовпцями знизу вгору і зліва на право,
5. за стовпцями зверху вниз і зліва на право,
6. по рядках зліва на право і від низу до верху,
7. за стовпцями знизу вгору і справа на ліво,
8. по рядках з права на ліво і зверху вниз;

Якщо ж налаштування відключено, то використовується «стандартний цикл» за стовпцями зверху вниз і зліва на право.

**4.Розмір мікроорганізму**

Ця функція впливає тільки на візуальну частину проекту: значення на trackBar означає розмір сторони квадрата в пікселях, що виводиться на екран.

РОЗДІЛ2 Графіки

Щоб не пересилати великі масиви інформації через форми через функції та змінні(як я це робив з налаштуваннями) для побудови графіків я вирішив використовувати .txt файл. Тут при кожному новому натиску кнопки «Графіки» txt-файл очищується і заповнюється заново.

**Спосіб заповнення txt-файла**

Коли кнопку «Графіки» натиснута в першу чергу txt-файл очищується. Потім кожною наступною строчкою заповнюємо масиви кількості мікроорганізмів на полі, концентрацію та конфігурацію наступним чином:

System.IO.StreamWriter file = new System.IO.StreamWriter(path, false);//очищаємо

file.WriteLine("");

file.Close();

file = new StreamWriter(path);

for (int i = 0; i < abc; i++)

file.Write(n[i] + " ");

file.Write(n[abc]);

**Побудова графіків**

Всі три види графіків будуються через елемент управління Chart:

1. Графік залежності кількості М від номеру ітерації (Рис. 4)
2. Графік концентрації М від номеру ітерації (Рис. 5)
3. Графік співпадаючих координат М з координатами минулої ітерації (%) (Рис. 6)

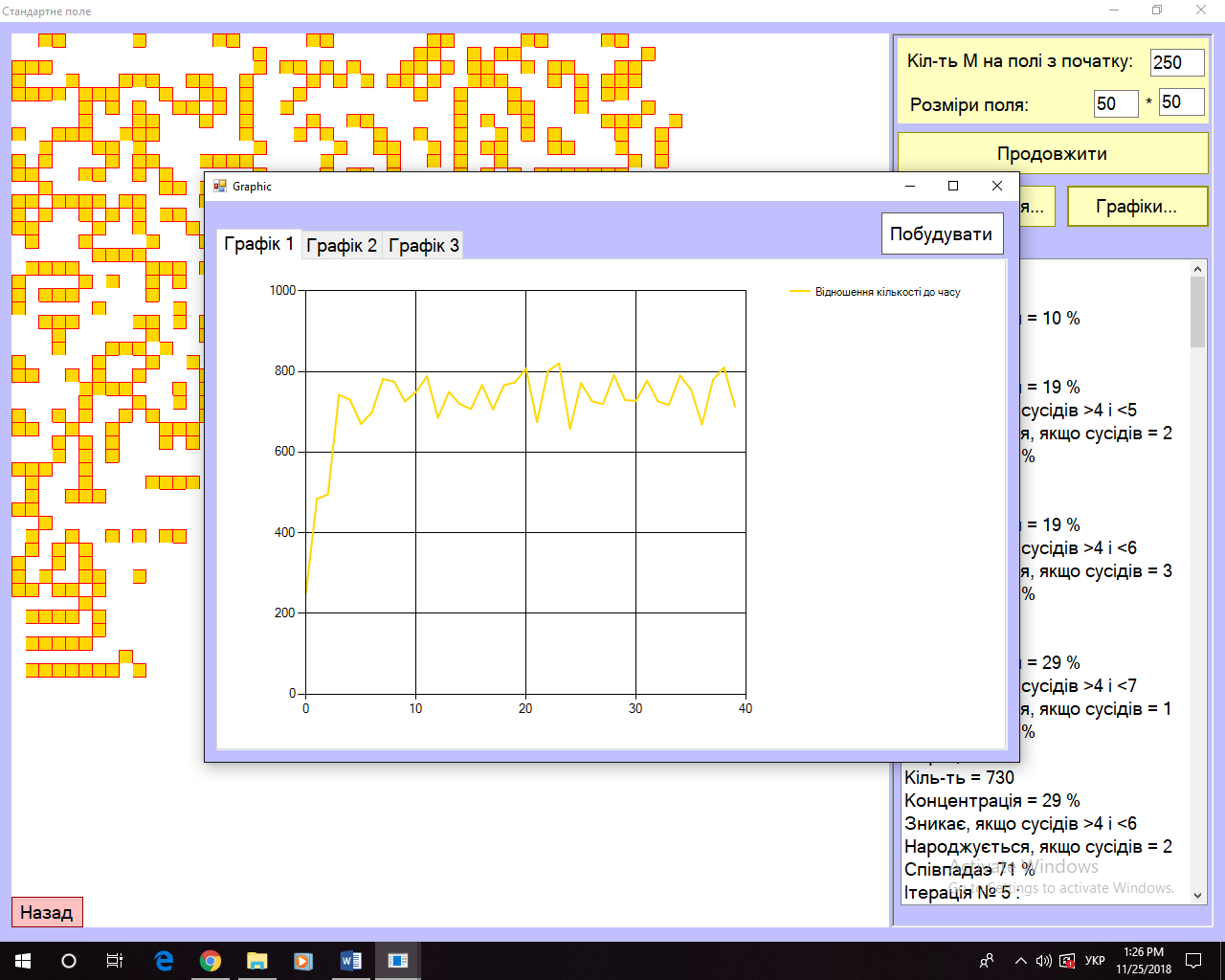


Рисунок 4. Графік кількості М від номеру ітерації

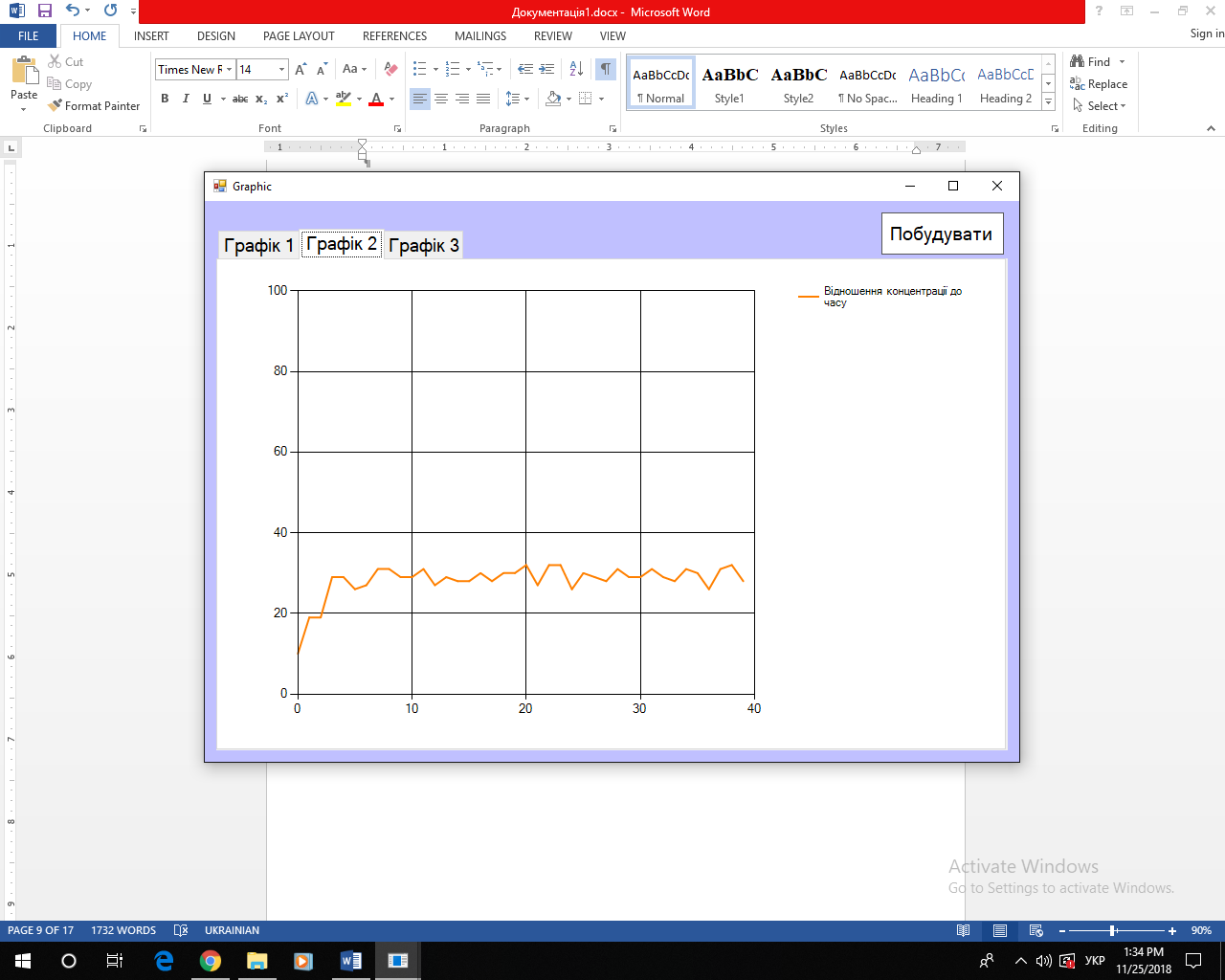


Рисунок 5. Графік концентрації М до номеру ітерації

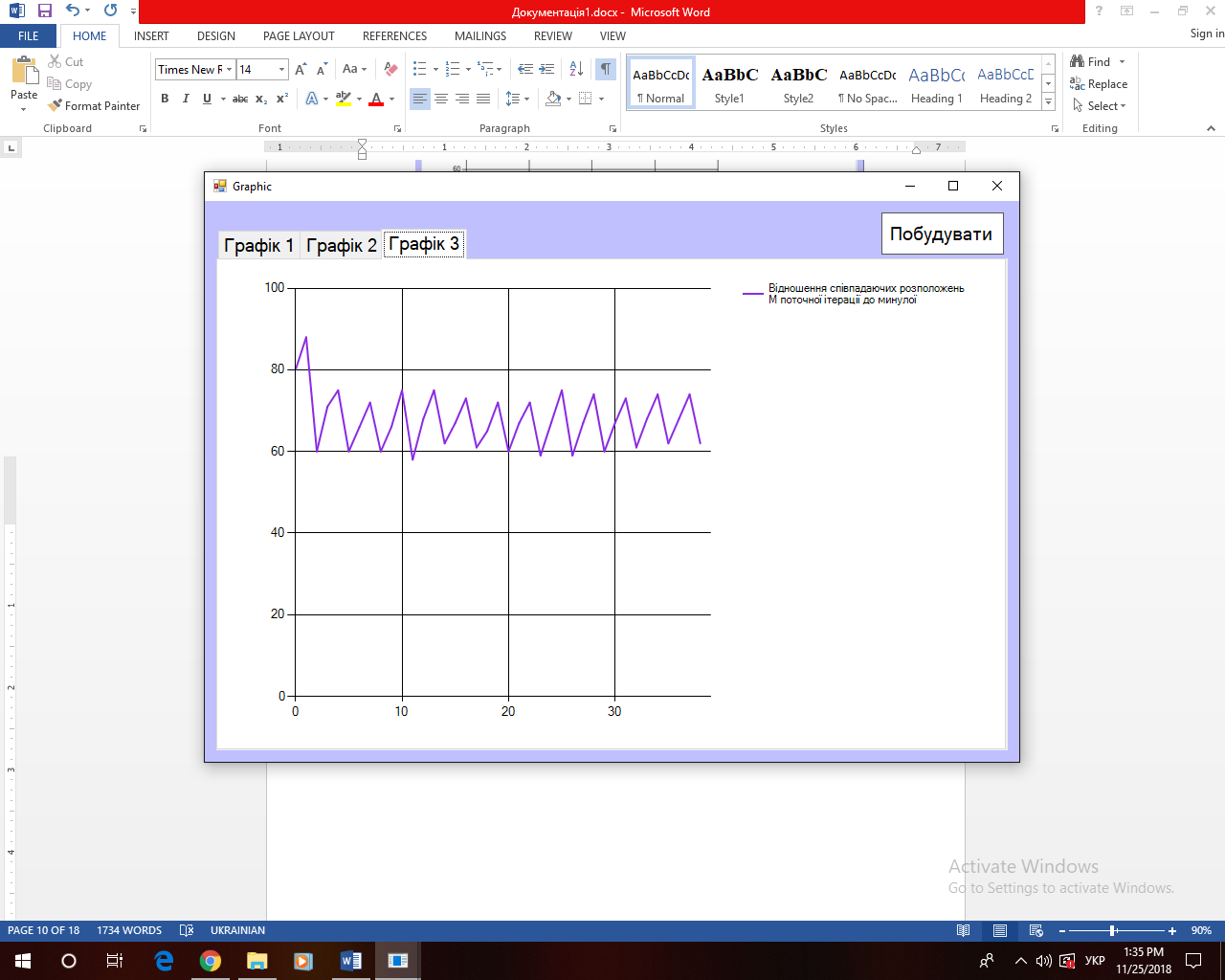


Рисунок 6. Графік зміни структури колонії мікроорганізму

РОЗДІЛ3 Стандартна колонія мікроорганізмів

Головною метою цього виду поля – це порівняння. Оскільки, змінюючи правила гри треба слідкувати і за змінами колонії мікроорганізмів. Також цей вид колонії є основним, на якому побудовані інші КМ. Тобто у кожній іншій КМ присутні всі деталі цього виду КМ, але із доповненнями.

У цій КМ впродовж усіх циклів життя підраховуються наступні значення:

1. Кількість мікроорганізмів (М) за кожну ітерацію;
2. Концентрація М на полі за кожну ітерацію(у%);
3. Співпадіння конфігурації М з конфігурацією попередньої ітерації(у%).

При натисканні кнопки «Розпочати» включається основний таймер, де спочатку послідовно змінюються умови виживання, потім проходить цикли життя. Тут перевіряється координата на присутність мікроорганізму: якщо ця клітина пуста, тоді перевіряється кількість сусідів на народжуваність М, або якщо в клітині є М, тоді перевіряється кількість сусідів на виживання:

int k = num\_of\_neigh(i, j);

if (a[i, j])//якщо в цій координаті є М, то перевіряємо кількість сусідів

{

if (k > q2 - 1 && k < q3 + 1)//

{

g.FillRectangle(Brushes.Gold, i \* z, j \* z, z, z);

g.DrawRectangle(Pens.Red, i \* z, j \* z, z, z);

}

else

{

a[i, j] = false;

g.FillRectangle(Brushes.White, i \* z, j \* z, z + 1, z + 1);

}

}

//якщо тут (в цій координаті) нічого нема і кількість сусідів = q1

//и якщо налаштування дозволяють, то тут з’являється новий М

birth(k, i, j, r);

Після цього фіксуються та перевіряються всі значення (кількість, концентрація, конфігурація)

Також всі зміни та розрахунки, які відбуваються виводяться у listBox під назвою «Підсумки».

Поточна кількість М на полі підраховується за допомогою лічильника та циклу; В залежності від поточної кількості М виводиться концентрація:

concentration[abc] = Convert.ToDouble(n[abc] \* 100 / (h \* w));

Далі вираховується, яке відношення клітинок поля, що не змінили свою наповненість до тих, що її змінили:

//перевіряємо конфігурацію

num\_of\_same\_m = compare(a\_g, a, h, w);

for (int i = 0; i < w; i++)

for (int j = 0; j < h; j++)

{

a\_g[i, j] = a[i, j];

}

configuration[abc - 1] = ((100 \* num\_of\_same\_m) / (w \* h));

listBox1.Items.Add("Співпадаэ " + configuration[abc-1].ToString() + " %");

num\_of\_same\_m = 0;

РОЗДІЛ4 Колонія мікроорганізмів з часовими полями

У цій КМ досліджується порівняння життя мікроорганізмів в залежності від початкових умов виживання і початкового розташування. А також від частоти зміни поля. Поле розділене на дві умовні частини навпіл по вертикалі. Кожна із частин має свій час зміни умов. Отже і для кожного із під полів є свій окремий таймер. У кожному із таймерів цикли життя відбуваються незалежно одне від одного, окрім змінної, яка відповідає за розташування мікроорганізмів.

Оскільки у обох таймерів відбуваються дії над однією і тією ж змінною, то навіть якщо відношення інтервалів таймерів є не ціле число, то таймери будуть змінюватися по черзі.

Також щоб дослідити залежність КМ від розташування при кожному новому початку гри М розташовуються у шаховому порядку. Це дозволяє вислідити поведінку колонії за однакових умов. Оскільки при шаховій розстановці перше часове поле розставляється з першої клітинки, а друге – з другої:

bool bl = true;

for (int i = 0; i < h; i++, bl = !bl)

for (int j = Convert.ToInt32(bl); j < w; j += 2)

{

a[i, j] = true;

g.FillRectangle(Brushes.Gold, i \* z, j \* z, z, z);

g.DrawRectangle(Pens.Red, i \* z, j \* z, z, z);

}

Візуального розділенням часових полів є

РОЗДІЛ5 Колонія мікроорганізмів з різною частотою змін умов та зміни покоління

Для зміни умов виживання урізний інтервал часу відносно ітерацій проходу перевірки колонії мікроорганізмів я зробив новий таймер(timer2). У цьому таймері у введений інтервал часу послідовно змінюються умови виживання в залежності від налаштувань:

if (q1 <= bmax - 1)

q1++;

else

q1 = bmin;

//q2 Є [ trackBar5 ; (trackBar5+trackBar4)/2 )

if (q2 < ((vmin + vmax) / 2) - 1)

q2++;

else

q2 = vmin;

//q3 Є [(trackBar5+trackBar4)/2 +1; trackBar4]

if (q3 < vmax)

q3++;

else

q3 = ((vmin + vmax) / 2) + 1;

if (a1 == false)

{

//f Є [1 ; 10]

if (f <= 9)

{

f++;

f = id\_limit(f, false);

}

else

{

f = 1;

f = id\_limit(f, false);

}

}

if (b1 == false)

{

//r Є [1 ; 10]

if (r <= 9)

{

r++;

r = id\_limit(r, true);

}

else

{

r = 1;

r = id\_limit(r, true);

}

}

Я дослідив випадки коли за один прохід циклу життя умови виживання змінюються багато разів. Тоді мікроорганізми змінюються майже кожний наступний з різними умовами. Тоді структура М виглядає хаотично і майже не спостерігається ніяких структур.

Також я дослідив обернений випадок, коли умови змінюються дуже повільно. У цьому випадку мікроорганізми змінюють свою конфігурацію під умови і майже повністю залишаються з постійною структурою.

ВИСНОВКИ

В продовж роботи я навчився працювати з передаванням інформації з основної форми на похідну та навпаки.

В цій роботі вперше було досліджено зміни умов виживання в залежності від часу. У відповідності із змінами умов змінюється і структура мікроорганізмів. З результатів графіків можна сказати, що кількість, структура та концентрація змінюються періодично. Ця періодичність залежить від частоти зміни умов виживання.

Створено три види клітинного автомату з використанням періодичних змін умов виживання, де користувач може змінювати їх інтервали.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

* <https://uk.wikipedia.org/wiki/%D0%96%D0%B8%D1%82%D1%82%D1%8F_(%D0%B3%D1%80%D0%B0)>
* <http://www.michurin.net/online-tools/life-game.html>
* http://vlad-utenkov.narod.ru/personal2/informat/km/1\_ka/1\_ka.htm

УМОВНІ ПОЗНАЧЕННЯ

КМ – Колонія Мікроорганізмів;

М – мікроорганізм або мікроорганізми;

Цикл життя – це прохід однієї ітерації головного таймера у будь-якому виді КМ;

Часове поле – це ділянка одного із вертикальних полів в КМ з часовими полями.